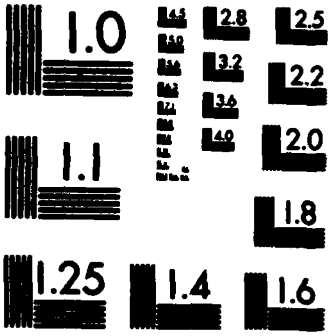
AD-A181 595 FINAL REPORT FOR CONTRACT NUMBER N00014-85-C-2513(U) KM 1/1
SCIENCES ARLINGTON VA 1987 N00014-85-C-2513

UNCLASSIFIED F/G 20/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

71

K M SCIENCES

FINAL REPORT

This is the final report of work performed by KM Sciences for the U.S. Naval Research Laboratory under contract N00014-85-C~~02513~~.
2513

DTIC
COPY
INSPECTED
6

This document has been approved for public release and sale; its distribution is unlimited.

Accesion For

NTIS	CRA&I	☒
DTIC	TAB	[ ]
Unannounced		[ ]
Justification		

By ltr. on file
Distribution /

Availability Codes

Dist	Avail and / or Special
A-1	

INTRODUCTION

During the period 13 September 1985 to 31 March 1987, KM Sciences furnished support to the radiation effects programs of the Radiation-Matter Interactions Branch of the Naval Research Laboratory in the areas of collecting, manipulating, analyzing, and displaying experimental data, in writing, modifying, converting, and extending computer codes for modeling physical phenomena and in graphical presentation of experimental data and results of computations. Two major tasks were supported: (1) computer simulation of the effects of particle beams on materials and (2) experimental measurements of the effects of particle beams on materials.

Keywords: MARPOP Computer program; MARPOP (Marlow Post Processor).

ACCOMPLISHMENTS

COMPUTER SIMULATION:

MARPOP, a computer program (written for the TI-ASC computer) that processes output from the MARLOW cascade simulation code, was converted to run on NRL's Cray computer. A new output routine for the Cray version of MARLOW was written to produce an output data file that serves as input to MARPOP. MARPOP performs calculations and manipulations on the data from MARLOW and produces tables and graphs summarizing and displaying the MARLOW results. A number of production runs were made with the Cray versions of MARLOW and MARPOP using input data designed to simulate experiments in which materials were irradiated with particle beams. In an evolutionary process, the experimental data and the results of the computer simulations led to frequent changes in the MARPOP processing and output specifications, requiring numerous changes in the MARPOP code. The results of these calculations were incorporated in a paper, "Cascade Simulation of the Crystal Orientation Dependence of Sputtering and Lattice Damage of Single Crystal Copper by Irradiation with 100 keV Copper Ions", by G.P.Mueller, M.Rosen, W.A.Fraser (KM Sciences), J.A.Sprague, P.R.Malmberg, J.M.Lambert, P.A.Treado, and G.W.Reynolds, published in Nuclear Instruments and Methods in Physics Research B18(1987)360-364. A source code listing of the most recent version of MARPOP is enclosed.

PARTICLE BEAM EXPERIMENTS:

Computer support was provided for five series of experiments using the NRL Linear Accelerator to study the effects of particle beams on materials. The support consisted of computer data acquisition, data manipulation and calculations, and graphical and tabular display of results.

OTHER:

A hands-on guide with sample command sequences was developed to introduce users to graphics software installed on the Condensed Matter and Radiation Sciences Division VAX computer. This enabled members of the Radiation-Matter Interactions Branch to produce many types of data plots with less than an hour of self-training using the guide.

Hardware and software specifications were developed for a personal-computer-based pilot system for storage and retrieval of experimental data and results of theoretical calculations of effects of charged particle beams and deposition of energy in materials. After the personal computer systems were delivered, they were set up and software was installed.

```
      PROGRAM MARPOP
C
C  VERSION 3.1                     22 MAY 1986
C
C  Programmer: W.A.FRASER       KM Sciences
C
C  MARPOP (MARlow POst-Processor) reads Cray program MARLOW 'POPDAT'
C  output file (data for each particle exiting from the target for zero
C  surface binding energy) and recalculates the yield, energy and
C  directional distributions for chosen surface binding energies.
C
C  THIS VERSION OF MARPOP PROCESSES OUTPUT FILES FROM MARLOW VERSION OF
C  18 (+ OR -1) FEBRUARY 1982, ADDING TABLES OF MAXIMUM PARTICLE DEPTH,
C  AND NUMBERS AND LENGTHS OF COLLISION SEQUENCES TO THE OUTPUT THAT WAS
C  PRODUCED BY MARPOP VERSIONS 1.3 AND 1.4.  THE "18" FEBRUARY MARLOW
C  VERSION TRUNCATES MAXIMUM DEPTH VALUES TO THE NEXT LOWER 0.1 ALAT(1)
C  UNIT.   DATA FROM UP TO 10 MARLOW OUTPUT FILES CAN BE COMBINED.   THE
C  NUMBER OF PRIMARY PARTICLES FOR WHICH DATA WILL BE PROCESSED FROM A
C  FILE CAN BE SET TO LESS THAN MAXRUN.   SURFACE BINDING ENERGY FOR
C  PRIMARY PARTICLES IS 0.0.
C
C                *****      Input Parameter Records       *****
C                *****   read from logical unit LUSPEC    *****
C
C  Up to 20 parameter sets may be input.   Each set consists of the
C  following:
C
C  Record                                                         Format
C
C    1    Title - may contain any readable characters              A80
C         (maximum = 80 characters)
C
C    2    NFILES                                                  10X,I2
C         Number of MARLOW 'POPDAT' files to be processed
C         (maximum = 10).
C
C    3    LIMRUN(NFILES)                                          10X,10I5
C         NFILES values of LIMRUN, the number of primary
C         particles for which data are to be processed from
C         each 'POPDAT' file.   The first value on record 3
C         will be used for file 'FT17', the next for 'FT18',
C         etc.   If LIMRUN(i)=0, the corresponding FTnn file
C         will be processed to the end (MAXRUN primaries).
C         If LIMRUN(i)>0, then data from the corresponding
C         FTnn file will be processed until data produced by
C         primary particle LIMRUN+1 are encountered.
C
C   4,5   SBND(1-10)                                              10X,5E10.0
C         Surface binding energies (e.v.) for up to 10
C         particle types.   MARLOWE Version 12 provides for
C         only 5 particle types.   Ten types were permitted
C         in Version 11.   Both records 4 and 5 must be
C         present (to process MARLOW Version 12 output there
C         will be no values on record 5).
C
C   6,7   WIDTH(1-10)                                             10X,5E10.0
C         Widths (e.v.) of channels for binning the energies
C         of up to 10 types of secondary particles.   MARLOWE
C         Version 12 provides for only 5 particle types.   Ten
C         types were permitted in Version 11.   Both records
```

```
C          6 and 7 must be present (to process MARLOW Version
C          12 output there will be no values on record 7).
C
C     8    WIDTH(11)                                         10X,E10.0
C          Width (e.v.) of channels for binning the energies
C          of primary particles.
C
C     9    DEPBIN, DEPMAX                                    10X,2E10.0
C          DEPBIN = bin width (in units of ALAT(1)) for
C                   depth distribution tables.
C          DEPMAX = bin width (in units of ALAT(1)) for
C                   maximum depth distribution tables.
C
C    10    NMUBIN, CHMU                                      10X,I2,1X,A2
C          NMUBIN = number of mu (cosine polar angle) to use
C                   for binning particles.   Present maximum
C                   is 20, limited by processing and output
C                   routines, but input routine provides for
C                   40 bins.   Arrays in other routines and
C                   output formats must be modified before
C                   40 bins can be used.
C          CHMU   = data type specifier.   CHMU='MU' if bin
C                   boundary values on record(s) 11 are in
C                   units of mu (cosine polar angle).  Leave
C                   CHMU blank if boundary values are in
C                   degrees.
C
C 11A,11B,etc. QMU(0 to NMUBIN) or POLDEG(0 to NMUBIN)       10X,7F10.0
C          As many records as are necessary for
C          NMUBIN+1 values of bin boundaries for binning
C          polar angle (theta) distribution of particles.
C          If CHMU (record 10) is 'MU', bin boundaries must
C          be cosines.   If CHMU is anything else, bin
C          boundaries must be in degrees.
C
C    12    NAZBIN                                           10X,I2
C          Number of azimuthal angle (phi) bins to use for
C          binning particles.   Present maximum is 20,
C          limited by processing and output routines, but
C          input routine provides for 40 bins.   Arrays in
C          other routines and output formats must be modified
C          before 40 bins can be used.
C
C 13A,13B,etc. AZMDEG(0 to NAZBIN)                           10X,7F10.0
C          As many records as are necessary for NAZBIN+1
C          values of bin boundaries for binning azimuthal
C          angle (phi) distribution of particles.   Values
C          must be in degrees.
C
C    14    XNORML                                            10X,F10.0
C          Factor (multiplier) to be used in normalizing
C          values for ejected particle direction distribution
C          table.   See writeup describing normalization of
C          array NANGLE in subroutine SUMRYZ.
C
C    15    OUTPUT CONTROL SWITCHES                           10X,A70
C          To eliminate undesired output, one or more of the
C          following codes (separated by commas or spaces)
C          may appear in columns 11-80, in any order.
C             FPRIM ("Front PRIMaries") suppresses output for
```

```
C                 reflected primaries.
C           RPRIM ("Rear PRIMaries") suppresses output for
C                 transmitted primaries.
C           FTARG ("Front TARGet") suppresses output for
C                 front sputtered target atoms.
C           RTARG ("Rear TARGet") suppresses output for
C                 rear sputtered target atoms.
C           INFO  causes file 'INFO' not to be included at
C                 the end of the print file.   The INFO file
C                 may be saved or disposed of just as any
C                 other file.
C        The record must be present so if all output is
C        desired, columns 11-80 should be blank.
C
C                     *****   OTHER INPUT   *****
C
C Data files (named 'POPDAT') written by program MARLOW from routine
C EXTRAI.  From 1 to 10 POPDAT files may be processed in a single
C MARPOP run.  All POPDAT files are processed for each MARPOP input
C parameter set described above.  POPDAT files are read from logical
C units 17,18,....,25,26, and must be assigned the names FT17, FT18,
C etc., up to FT26.
C ETC.).
C
C
C                 *****   VARIABLE DECLARATIONS   *****
C
C             Carriage control characters
      CHARACTER*1 CHPLUS, CHWUN
C             Output switches
      CHARACTER*5 FPRIM, RPRIM, FTARG, RTARG, INFO
      CHARACTER*8 CHKODE
      CHARACTER*80 CHKARD
      CHARACTER*130 CHLINE
      INTEGER   I, ISET, KODE, LDATE, LIMRUN, LTIME, LUNIT,
     1           MXFYLS, NAZBIN, NFILES
      REAL      DEPBIN, DEPMAX, SBND, WIDTH
C
C             PRIMARY PARTICLE PROCESSING LIMITS READ FROM CARD(S) 3
      COMMON /LIMITS/ LIMRUN(10)
C             Output switches
      COMMON /OUTSWT/ FPRIM, RPRIM, FTARG, RTARG, INFO
      COMMON /READIN/ SBND(11), WIDTH(11), DEPBIN, DEPMAX
C             I/O logical units set in subprogram BLOKAA
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C      NEXT STATEMENT FORCES LINKING OF BLOCK DATA SUBPROGRAM THAT SETS
C      LOGICAL UNIT NUMBERS.
C
C      Next statement forces linking of block data subprogram that sets
C      constant values for I/O logical units, etc.
      EXTERNAL BLOKAA
C
      DATA CHPLUS /'+'/, CHWUN /'1'/
C             Maximum number of POPDAT files that may be processed
      DATA MXFYLS /10/
C
  100 FORMAT(A80)
  101 FORMAT(10X,5E10.0)
  102 FORMAT(10X,2E10.0,I5)
  103 FORMAT(10X,I2)
  104 FORMAT(10X,10I5)
  105 FORMAT(A130)
  200 FORMAT(A1/'0',7X,'PROGRAM MARPOP  -  MARLOW POST PROCESSOR',15X,
     1   'VERSION 3.1',5X,'OF 22 MAY 1986')
  201 FORMAT(' ',7X,'*** UNEXPECTED END-OF-FILE, INPUT DATA CARDS')
  202 FORMAT(' ',7X,'*** READ ERROR, INPUT DATA CARDS')
  203 FORMAT(' ',7X,'PROCESSING',I4,2X,'POPDAT FILES')
  204 FORMAT(' ',7X,'PROGRAM MARPOP FINISHED')
  205 FORMAT(/'0',27X,A80/)
  206 FORMAT(' ',7X,'*** UNEXPECTED END-OF-FILE, UNIT ',I4)
  207 FORMAT(' ',7X,'*** READ OR DATA ERROR, UNIT ',I4)
  208 FORMAT(' ',7X,
     1     '*** MAXIMUM NUMBER OF RECORDS PROCESSED FROM UNIT ',I4)
  209 FORMAT(' ',7X,'*** END-OF-FILE READ, INPUT DATA CARDS')
  210 FORMAT(' ',7X,'UNIT ',I4,5X,'MAXRUN PRIMARIES TO BE PROCESSED')
  211 FORMAT(' ',7X,'UNIT ',I4,I10,2X,'PRIMARIES TO BE PROCESSED')
  212 FORMAT('0',7X,'DATE',1X,A8,5X,'TIME',1X,A8)
  213 FORMAT('1CONTENTS OF FILE INFO')
  214 FORMAT(A130)
  215 FORMAT(' LAST RECORD OF FILE INFO')
C
      WRITE (LUPRT,200) CHPLUS
      CALL DATE (LDATE)
      CALL CLOCK (LTIME)
      OPEN (UNIT=LUINFO,FILE='INFO',STATUS='NEW')
      WRITE (LUPRT,212) LDATE, LTIME
      WRITE (LUINFO,200) CHPLUS
      WRITE (LUINFO,212) LDATE, LTIME
C
C               On each pass through DO 4000 ISET loop, one set of
C               parameter records is read from logical unit LUSPEC and
C               processed.
C
      DO 4000 ISET=1,20
C                                                        *** Record 1 ***
         READ (LUSPEC,100,END=8000,ERR=10000) CHKARD
         IF (ISET.GT.1) WRITE (LUPRT,200) CHWUN
         WRITE (LUPRT,205) CHKARD
         WRITE (LUINFO,205) CHKARD
C                                                        *** Record 2 ***
         READ (LUSPEC,103,END=9000,ERR=10000) NFILES
         IF (NFILES.GT.MXFYLS) NFILES=MXFYLS
         WRITE (LUPRT,203) NFILES
C                                                        *** Record 3 ***
         READ (LUSPEC,104,END=9000,ERR=10000) (LIMRUN(I),I=1,NFILES)
C                                                     *** Records 4,5 ***
         READ (LUSPEC,101,END=9000,ERR=10000) (SBND(I),I=1,10)
C                                                   *** Records 6,7,8 ***
         READ (LUSPEC,101,END=9000,ERR=10000) WIDTH
C                                                        *** Record 9 ***
         READ (LUSPEC,102,END=9000,ERR=10000) DEPBIN, DEPMAX
C
C               Remaining parameter records are read from LUSPEC in
C               subroutine INPTB.
C
         CALL INPTB (CHKODE)
         IF (CHKODE.EQ.'EOF     ') THEN
            GO TO 9000
         ELSE IF (CHKODE.EQ.'ERROR   ') THEN
            GO TO 10000
         END IF
         CALL INITLZ
C
C               On each pass through DO 3000 JFYL loop one MARLOW
C               'POPDAT' file is read and particle data tabulated.
C
         DO 3000 JFYL=1,NFILES
            LUDAT = 16 + JFYL
            IF (ISET.GT.1) REWIND LUDAT
            IF (LIMRUN(JFYL).LE.0) THEN
               WRITE (LUPRT,210) LUDAT
            ELSE
               WRITE (LUPRT,211) LUDAT, LIMRUN(JFYL)
            END IF
C
C               Read first record from logical unit LUDAT
C
            CALL INPTA (JFYL, LUDAT, KODE)
C
C                    Values of KODE returned by INPTA:
C                         1 = Data read OK
C                         2 = End-of-file read, no FTxx file
C                         3 = Read error, file FTxx
C
            GO TO (1000, 5000, 7000) KODE
 1000       CONTINUE
            CALL PRTCLS (ISET, JFYL, LUDAT, KODE)
C                                   VALUES OF KODE RETURNED BY PRTCLS:
C                                    1  RECORDS READ AND PROCESSED OK
C                                    2  UNEXPECTED END-OF-FILE, NO DATA
C                                    3  UNEXPECTED END-OF-FILE, PARTIAL
C                                       DATA
C                                    4  READ ERROR, NO DATA
C                                    5  READ ERROR, PARTIAL DATA
C                                    6  MAXIMUM NUMBER OF RECORDS READ
C                                       AND PROCESSED
            GO TO (3000, 5000, 6000, 7000, 10500, 2000) KODE
 2000       CONTINUE
            WRITE (LUPRT,208) LUDAT
 3000    CONTINUE
      CALL SUMRYZ
C                                   PRINT TABLES AND HISTOGRAMS
         CALL OUPTAA
 4000 CONTINUE
      GO TO 12000
 5000 CONTINUE
      WRITE (LUPRT,206) LUDAT
      IF (I.GT.1) GO TO 11000
      GO TO 12000
 6000 CONTINUE
      WRITE (LUPRT,206) LUDAT
      GO TO 11000
 7000 CONTINUE
      WRITE (LUPRT,207) LUDAT
      IF (I.GT.1) GO TO 11000
      GO TO 12000
 8000 CONTINUE
      IF (ISET.LE.1) GO TO 9000
         WRITE (LUPRT,209)
         GO TO 12000
 9000 CONTINUE
      WRITE (LUPRT,201)
      GO TO 12000
10000 CONTINUE
      WRITE (LUPRT,202)
      GO TO 12000
10500 CONTINUE
      WRITE (LUPRT,207) LUDAT
11000 CONTINUE
C                                   PRINT TABLES AND HISTOGRAMS
      CALL OUPTAA
12000 CONTINUE
      WRITE (LUINFO,215)
      ENDFILE LUINFO
      WRITE (LUPRT,204)
      IF (INFO.EQ.'INFO ') THEN
         REWIND LUINFO
         WRITE (LUPRT,213)
         DO 13000 I=1,600
            READ (LUINFO,105,END=14000) CHLINE
            WRITE (LUPRT,214) CHLINE
13000    CONTINUE
      END IF
14000 CONTINUE
C
C         Next statement is only normal program execution stop
C
      STOP
      END
      SUBROUTINE ANGPLT (KTYPE, KSURF, AVYELD, ERROR)
C
C  13 MARCH 1986
C
C  ANGPLT MAKES LINE PRINTER HISTOGRAM PLOT OF ANGULAR DISTRIBUTION
C
C                  *****   CALLING PARAMETERS   *****
C
C  KTYPE  - (INTEGER, PASSED) PARTICLE TYPE NUMBER (1-10 FOR SECONDARY,
C           11 FOR PRIMARY PARTICLES)
C  KSURF  - (INTEGER, PASSED) TARGET SURFACE ID (1=FRONT, 2=BACK)
C  AVYELD - (REAL, PASSED) MEAN SECONDARY PARTICLE YIELD PER PRIMARY
C  ERROR  - (REAL, PASSED) STANDARD DEVIATION OF MEAN YIELD (AVYELD)
C
C  THE PLOT IS SET UP IN ARRAY LYNPLT, 81 COLUMNS (1=LEFTMOST) BY 42
C  LINES (LINE 2 IS TOP OF PLOT FRAME, LINE 1 IS PRINTED ABOVE THE
C  FRAME).  IF COSINE BIN 20 (NPLOT(20,KTYPE,KSURF)) IS GREATER THAN
C  ZERO, THE HEIGHT OF THE RIGHTMOST HISTOGRAM BAR = 1.0, AND THE OTHER
C  BAR HEIGHTS ARE PROPORTIONED TO IT.   IF NPLOT(20,KTYPE,KSURF) = 0,
C  THE LARGEST BIN IS FOUND AND ASSIGNED A HEIGHT OF 1.0, AND THE OTHER
C  BARS ARE PROPORTIONED TO THAT ONE.   IF A BAR HAS A CALCULATED HEIGHT
C  GREATER THAN 1.0, THE TOP OF THE BAR EXTENDS JUST ABOVE THE PLOT AND
C  THE CALCULATED HEIGHT (ENCODED INTO ARRAY IXCEED) IS PRINTED ABOVE
C  THE TOP OF THE BAR.
C
C                 *****   VARIABLE DECLARATIONS   *****
C
C
C             PLOT AXIS AND HEADING LABELS
      DIMENSION LABLAX(11), LABSRF(4,2)
C             STORAGE FOR PLOT CHARACTERS
```

```
      DIMENSION IXCEED(20), LYNPLT(81,42), LYNOUT(324,10), LEG1(24),
     1          LEG2(24), LEG3(24), LEG4(24)
C             TEMPORARY STORAGE FOR PLOT LEGEND LINES
      DIMENSION LEGEND(3)
C
      COMMON /BINS/ POLDEG(0:40), GMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
C             ADUMMY = ARRAY NOT USED IN THIS SUBPROGRAM
C             LTYPE  = PARTICLE ID (LITERAL)
      COMMON /FSTREC/ ADUMMY(48), LTYPE(10), DDUMMY(70)
C             SURFACE BINDING ENERGIES
      COMMON /READIN/ SBND(11)
C             ENERGY (NENERG), YIELD (NYIELD), DEPTH (NDEPTH), AND
C                      ANGULAR (NANGLE) DISTRIBUTION TABLES.
C             KSUM, KSUMSQ = SUM AND SUM OF SQUARES OF YIELD FOR
C                      CALCULATING MEAN YIELD AND STANDARD DEVIATION.
C             NPRYMS = TOTAL NUMBER OF PRIMARY PARTICLES.
      COMMON /TABLES/ NENERG(100,11,2), NYIELD(21,10,2),
     1                NDEPTH(21,10,2), NRMLZD(21,21,11,2), FACNRM(11,2),
     2                NANGLE(21,21,11,2), KSUM(10,2), KSUMSQ(10,2),
     3                NPRYMS
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C
      EQUIVALENCE (LEG1(1),LYNPLT(4,4)), (LEG2(1),LYNPLT(4,5)),
     1            (LEG3(1),LYNPLT(4,6)), (LEG4(1),LYNPLT(4,7)),
     2            (LYNOUT(1,1),LYNPLT(1,2))
      SAVE
C
      DATA LABLAX /'0.0 ', '0.1 ', '0.2 ', '0.3 ', '0.4 ', '0.5 ',
     1             '0.6 ', '0.7 ', '0.8 ', '0.9 ', '1.0 '/
      DATA LABSRF /'  RE', 'FLEC', '  FR', 'ONT ', 'TRAN', 'SMIT',
     1             '   R', 'EAR '/
C             NUMBER OF CHARACTERS IN ARRAY LEGEND
      DATA NLEG /28/
C             SYMBOLS FOR PLOT
      DATA IBLANK /'    '/, IDASH /'-'/, IPLUS /'+'/
      DATA MXAZM /21/
C
  201 FORMAT(1H1////1H0,34X,'ANGULAR DISTRIBUTION',3X,'-',1X,2A4,
     1    'SPUTTERED',2X,A2,I3,2X,'ATOMS')
  202 FORMAT(1H1////1H0,33X,'ANGULAR DISTRIBUTION',3X,'-',3X,2A4,
     1    'TED PRIMARY PARTICLES')
  203 FORMAT(1H ,24X,20A4)
  204 FORMAT((1H ,19X,A4,81A1,1X,A3/ 2(1H ,23X,81A1/), 1H ,23X,81A1))
  205 FORMAT(20X,21(3X,'I') / 22X,21(F4.1) /
     1  'O',49X,'Polar Angle (Degrees)'/)
  206 FORMAT(1H0,7X,'*** NO DATA FOR ANGULAR DISTRIBUTION PLOT'/ 1H1)
  207 FORMAT('O',19X,'1.0=',I8,2X,'Particles/dMu')
  401 FORMAT(F4.1)
  402 FORMAT('SURF.BINDG.EN.',1PE9.2,1X)
  403 FORMAT('PRIMARIES',5X,I7,3X)
  404 FORMAT('MEAN YIELD',1X,F10.1,3X)
  405 FORMAT('ERROR',6X,F10.1,3X)
C
C                                     PRINT HEADING
      IF (KTYPE.LT.11) PRINT 201, (LABSRF(I,KSURF),I=3,4), LTYPE(KTYPE),
     1  KTYPE
      IF (KTYPE.EQ.11) PRINT 202, (LABSRF(I,KSURF),I=1,2)
C                                     BLANK PLOT ARRAY, PUT IN GRID LINES
      DO 800 I=1,20
```

```
            IXCEED(I) = IBLANK
  800 CONTINUE
      DO 900 I=1,42
         DO 900 J=1,81
            LYNPLT(J,I) = IBLANK
  900 CONTINUE
      DO 1000 I=2,42,8
         DO 1000 J=1,81,2
            LYNPLT(J,I) = IDASH
 1000       CONTINUE
      DO 2000 J=2,42
         DO 2000 I=1,81,16
            LYNPLT(I,J) = IDASH
 2000       CONTINUE
C                                SET HISTOGRAM BAR HEIGHT REFERENCE
      IYTOP = NRMLZD(1,MXAZM,KTYPE,KSURF)
         DO 3000 I=NMUBIN,NMUBIN
            IF (NRMLZD(I,MXAZM,KTYPE,KSURF).GT.IYTOP)
     1         IYTOP=NRMLZD(I,MXAZM,KTYPE,KSURF)
 3000       CONTINUE
         IF (IYTOP.GT.0) GO TO 4000
            PRINT 206
            RETURN
C
 4000 CONTINUE
      YMAX = FLOAT(IYTOP)
C                                SET UP HISTOGRAM IN PLOT ARRAY
      LYNPLT(1,42) = IPLUS
      LAST = 42
C                                HISTOGRAM BAR FOR ONE OF THE 20
C                                COSINE BINS IN NRMLZD IS SET UP ON
C                                EACH PASS THROUGH DO 14000 I LOOP.
C                                I IS BIN SUBSCRIPT IN NRMLZD.
C                                I+1 IS CURRENT BAR IN ARRAY LYNPLT.
      DO 14000 I=1,NMUBIN
         NPLX40 = 40 * NRMLZD(I,MXAZM,KTYPE,KSURF)
         LINTMP = NPLX40 / IYTOP
         IF ((NPLX40-(LINTMP*IYTOP)).GT.0) LINTMP=LINTMP+1
         LINE = 42 - LINTMP
         IF (LINE.GT.1) GO TO 5000
            RATIO = FLOAT(NRMLZD(I,MXAZM,KTYPE,KSURF)) / YMAX
            ENCODE (4,401,IXCEED(I)) RATIO
            LINE = 1
 5000    CONTINUE
         JSTART = 2 + (4*(I-1))
         JSTOP = JSTART + 3
         DO 6000 J=JSTART,JSTOP
            LYNPLT(J,LINE) = IPLUS
 6000       CONTINUE
         IF (LINE.GE.41) GO TO 8000
            KSTART = LINE + 1
            DO 7000 K=KSTART,41
               DO 7000 J=JSTART,JSTOP
                  LYNPLT(J,K) = IBLANK
 7000             CONTINUE
 8000    CONTINUE
         IF (LINE-LAST) 9000, 14000, 11000
 9000       CONTINUE
            ICOLUM = JSTART
            DO 10000 J=LINE,LAST
```

```
               LYNPLT(ICOLUM,J) = IPLUS
10000          CONTINUE
            GO TO 13000
11000       CONTINUE
            ICOLUM = JSTART - 1
            DO 12000 J=LAST,LINE
               LYNPLT(ICOLUM,J) = IPLUS
12000          CONTINUE
13000    CONTINUE
         LAST = LINE
14000    CONTINUE
      DO 15000 J=LAST,42
         LYNPLT(81,J) = IPLUS
15000    CONTINUE
C                                   PUT LEGEND IN PLOT.   CIMOVE
C                                   EXPANDS LEGEND LINE FROM ARRAY
C                                   LEGEND INTO PLOT ARRAY.
      ENCODE (24,402,LEGEND(1)) SBND(KTYPE)
      CALL CIMOVE (LEGEND, LEG1)
      ENCODE (24,403,LEGEND(1)) NPRYMS
      CALL CIMOVE (LEGEND, LEG2)
C                                   OMIT OTHER LEGEND LINES FOR PRIMARY
C                                   PARTICLE
      IF (KTYPE.GT.10) GO TO 16000
         ENCODE (24,404,LEGEND(1)) AVYELD
         CALL CIMOVE (LEGEND, LEG3)
         ENCODE (24,405,LEGEND(1)) ERROR
         CALL CIMOVE (LEGEND, LEG4)
16000 CONTINUE
C                                   PRINT THE HISTOGRAM
      PRINT 204, IBLANK, (LYNPLT(I,1),I=1,81)
      PRINT 204, (LABLAX(12-I), (LYNOUT(J,I),J=1,81), LABLAX(12-I),
     1   (LYNOUT(K,I),K=82,324), I=1,10)
      PRINT 204, LABLAX(1), (LYNPLT(I,42),I=1,81)
      WRITE (LUPRT,205) (POLDEG(I),I=0,NMUBIN)
      WRITE (LUPRT,207) IYTOP
      RETURN
      END
      BLOCK DATA BLOKAA
C
C  12 MARCH 1986
C
C  BLOKAA sets values for some constants used in program MARPOP.
C
C        Array dimensions; values used mainly for loop indices.
C             MXSURF = Maximum number of target surfaces
C             MXTYPE = Maximum number of particle types.   Numbers 1-5
C                      are used for secondary particles (10 were
C                      permitted by MARLOWE Version 11).  Type 11 is
C                      primaries.
C             MXPOLR = Maximum number of values for boundaries of polar
C                      angle bins.
C             MXAZM  = Maximum number of values for boundaries of
C                      azimuthal angle bins.
      COMMON /MXDMNS/ MXSURF, MXTYPE, MXPOLR, MXAZM
C
C        I/O Logical unit assignments
C             LUSPEC = Input parameter file
C             LUPRT  = Output print file
C             LUINFO = Output "information" file
```

```
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C
      DATA MXSURF /2/, MXTYPE /11/, MXPOLR /21/, MXAZM /21/
      DATA LUSPEC /5/, LUPRT /6/, LUINFO /9/
C
      END
      SUBROUTINE CIMOVE (ISORCE, ITARGT)
C
C  24 JANUARY 1986
C
C  CIMOVE EXPANDS EIGHT CHARACTERS FROM EACH WORD OF ISORCE INTO
C  ARRAY ITARGT, ONE CHARACTER PER WORD, IN LEFTMOST BYTE, WITH REST
C  OF WORD FILLED WITH ASCII SPACES (40 OCTAL, 20 HEXADECIMAL).
C
      DIMENSION ISORCE(3), ITARGT(24)
C
      DO 1000 I=1,3
         DO 1000 J=1,8
            ITARGT(8*(I-1)+J) = X'0020202020202020' +
     1         AND(SHIFT(ISORCE(I),8*(J-1)),X'FF00000000000000')
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE FRONT (KTYPE, DEEP, DEPMAX, LA, KARMA)
C
C  17 JANUARY 1986
C
C  FRONT extracts and tabulates the following data for particles
C  escaping from the front surface: (a) number of atomic collisions,
C  (b) numbers of replacements in replacement sequences, and (c) maximum
C  distance from surface.   A second entry point, FRINIT, is used to
C  initialize the arrays in which data are tabulated.
C
C                     *****   CALLING PARAMETERS   *****
C
C  KTYPE     (integer,passed) is the particle type.   MARLOWE Version 11
C            provided up to 10 types.   Version 12 provides only 5 types.
C            Therefore, current permitted values of KTYPE are 1-5 and 11,
C            for up to 5 types of secondary particles (1-5) with a value
C            of 11 indicating a primary particle.
C
C  DEEP      (real,passed)  d + (100 * (10*z)), where d is MARLOWE
C            variable DEEP, and z is the maximum depth (in MARLOWE
C            ALAT(1) units, truncated to the next lowest 0.1 unit)
C            reached by the particle.
C
C  DEPMAX    (real,passed) Bin width (in MARLOWE ALAT(1) units) for
C            maximum depth distribution tables.
C
C  LA        (integer,passed)  m + 256*n, where m is MARLOWE variable LA
C            (a particle type identifier) and n is the number of atomic
C            collisions of the particle between its deepest point in the
C            slab and the escape (front) surface.
C
C  KARMA     (integer,passed)
C            k + 65536 * (a + 16*b + (16**2)*c + (16**3)*d + (16**4)*e)
C            where k is MARLOWE variable KARMA and a,b,c,d,e are the
C            numbers of replacements in each of five replacement
C            sequences.
C
```

```
C  This processing was grouped together and put in a subroutine to
C  minimize the modifications to subroutine CALCNS in MARPOP Version 2.0
C  for TI-ASC.
C
C  Data are tabulated in the arrays MXDEP, LENSEQ, and NCOLIS.   In
C  those arrays the dimension 11 represents type of atom, passed to this
C  routine in calling parameter KTYPE.   Only values 1-5 and 11 are used
C  by data from MARLOWE Version 12, so nearly half of each array is not
C  used (see description of KTYPE above.)
C
C  MXDEP   Maximum depth distribution table.  Depths are measured in
C          units of MARLOWE variable ALAT(1).   21 bins are provided for
C          each particle type.   MXDEP(1,i) is count of particles
C          reaching maximum depth less than 1 unit; MXDEP(2,i) counts
C          particles reaching depth of 1 unit, but less than 2 units;
C          ..... ; MXDEP(20,i) counts particles reaching 19 units but
C          less than 20 units; MXDEP(21,i) is count of particles
C          reaching 20 units or more.
C
C  LENSEQ  Length of replacement sequences table.   15 collisions in
C          sequence X 5 sequences X 11 particle types.
C
C  NCOLIS  Number of collisions from deepest point table.   First index
C          runs 1-21 for 1-20 collisions and .GT.20 in 21st bin.
C          Second index is for 11 particle types.
C
      INTEGER  I, KARMA, KARTMP, KCOLIS, KMXDEP, KOUNT, KTYPE, LA,
     1            LENSEQ, MXDEP, NCOLIS, NSEQ
      REAL      DEEP, DEPMAX
      DIMENSION NSEQ(5)
C
      COMMON /FRSURF/ MXDEP(21,11), LENSEQ(15,5,11), NCOLIS(21,11)
C
C            Next statement causes values of all variables to be saved
C            on exit from this subroutine.
      SAVE
C
C         *****   Extract and bin maximum depth data   *****
C
C            KMXDEP is first calculated as number of DEPMAX units,
C            then converted to bin index.
C            1.0E-8 in next statements prevents roundoff error
C
      KMXDEP = IFIX(((DEEP-AMOD(DEEP,100.0))/(1000.0*DEPMAX))+1.0E-8)
      IF (KMXDEP.GT.20) KMXDEP=20
      MXDEP(KMXDEP+1,KTYPE) = 1 + MXDEP(KMXDEP+1,KTYPE)
C
C            *****   Extract and bin collision data   *****
C
      KCOLIS = LA / 65536
      IF (KCOLIS.GT.20) KCOLIS=20
      NCOLIS(KCOLIS+1,KTYPE) = 1 + NCOLIS(KCOLIS+1,KTYPE)
C
C        *****   Extract and bin replacement sequence data   *****
C
      KARTMP = KARMA / 256
      KOUNT = 0
      DO 1000 I=1,5
         NSEQ(I) = MOD(KARTMP,16)
         IF (NSEQ(I).GT.0) KOUNT=KOUNT+1
         KARTMP = KARTMP / 16
 1000    CONTINUE
      DO 2000 I=1,5
         IF (NSEG(I).GT.0) LENSEG(NSEG(I),KOUNT,KTYPE) =
     1      1 + LENSEG(NSEG(I),KOUNT,KTYPE)
 2000    CONTINUE
      RETURN
C
C           *************************************************
C           *****                                       *****
C           *****   ENTRY POINT TO INITIALIZE ARRAYS    *****
C           *****                                       *****
C           *************************************************
C
      ENTRY FRINIT
C
      DO 3000 I=1,11
         DO 3000 J=1,21
            MXDEP(J,I) = 0
 3000 CONTINUE
      DO 4000 I=1,11
         DO 4000 J=1,21
            NCOLIS(J,I) = 0
 4000 CONTINUE
      DO 5000 I=1,11
         DO 5000 J=1,5
            DO 5000 K=1,15
               LENSEG(K,J,I) = 0
 5000 CONTINUE
      RETURN
      END
      SUBROUTINE INITLZ
C
C  12 MARCH 1986
C
      LOGICAL LOJDEG
      COMMON /BINS/ POLDEG(0:40), QMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
      COMMON /MORBNS/ XMUMID(20), PHIMID(40)
C
C         SAVE values.
      SAVE
C
      DATA MXBINS /40/, PI /3.141592653590/
C
      RADFAC = PI / 180.0
C
C                    *********************************
C                    *                               *
C                    *   THETA (POLAR ANGLE) BINS    *
C                    *                               *
C                    *********************************
C
C          Initialize polar angle bin boundaries: Non-zero POLDEG value
C          indicates input boundaries were in degrees and boundaries in
C          cosine (QMU) units are calculated in DO 3000 I loop.   If
C          POLDEG values are all zero, input boundaries were in mu units
C          and boundaries in degrees are calculated in DO 4000 I loop.
C
      LOJDEG = .FALSE.
      DO 1000 I=0,MXBINS
         IF (POLDEG(I).NE.0.0) THEN
            LOJDEG = .TRUE.
            GO TO 2000
         END IF
 1000 CONTINUE
 2000 CONTINUE
      IF (LOJDEG) THEN
         DO 3000 I=0,NMUBIN
            QMU(I) = COS(RADFAC*POLDEG(I))
 3000    CONTINUE
      ELSE
         DO 4000 I=0,NMUBIN
            POLDEG(I) = ACOS(QMU(I)) / RADFAC
 4000    CONTINUE
      END IF
C
C          Calculate width (DMU) and midpoint (XMUMID) of each theta bin
C
      DO 5000 I=1,NMUBIN
         DMU(I) = QMU(I) - QMU(I-1)
         XMUMID(I) = 0.5 * (QMU(I) + QMU(I-1))
 5000 CONTINUE
C
C                  ***********************************
C                  *                                 *
C                  *   PHI (AZIMUTHAL ANGLE) BINS    *
C                  *                                 *
C                  ***********************************
C
C          Calculate phi boundaries (AZMRAD) in radian, and width
C          (DAZBIN) and midpoint (PHIMID) of each phi bin.
C
      DO 6000 I=0,NAZBIN
         AZMRAD(I) = RADFAC * AZMDEG(I)
 6000 CONTINUE
      DO 7000 I=1,NAZBIN
         DAZBIN(I) = AZMRAD(I) - AZMRAD(I-1)
         PHIMID(I) = 0.5 * (AZMRAD(I) + AZMRAD(I-1))
 7000 CONTINUE
      RETURN
      END
      SUBROUTINE INPTA (ITER, LUDAT, KODE)
C
C  24 JANUARY 1986
C
C  INPTA READS FIRST RECORD WRITTEN BY PROGRAM MARLOW AND PRINTS MARLOW
C  RUN DATE AND TIME IN HEADING
C
C                 *****   CALLING PARAMETERS   *****
C
C  ITER (INTEGER, PASSED)  FILE IDENTIFIER
C  LUDAT (INTEGER, PASSED)  LOGICAL UNIT FROM WHICH CURRENT INPUT FILE
C        IS READ.
C  KODE (INTEGER, RETURNED)  VALUES OF KODE RETURNED BY INPTA:
C        1  DATA READ OK
C        2  End-of-file read, no file FTnn (nn=LUDAT)
C        3  Read error, file FTnn (nn=LUDAT)
C
      CHARACTER*8 LDATE, LTIME, JOBID(5), IDENT(15), OLDTYP, NEWTYP
      INTEGER  I, INREC, INTEMP, ITER, KODE, LTYPE, NTYPE
C
C             ARRAY INTO WHICH RECORD IS READ
      DIMENSION INTEMP(128)
      DIMENSION LTYPE(10)
C
      COMMON /FSTREC/ INREC(128)
C
      EQUIVALENCE (NTYPE, INREC(7)),  (LTYPE(1), INREC(49))
      EQUIVALENCE (MAXRUN, INTEMP(6))
      SAVE
C
  201 FORMAT(' ',7X,
     1    'PROCESSING OUTPUT FROM PROGRAM MARLOW EXECUTION OF',2(1X,A8)/
     2    '0',5A8,5X,'MAXRUN=',I10/' ',15A8)
  202 FORMAT('0',7X,'*** UNIT',2X,I4,5X,
     1    'NUMBER OF PARTICLE TYPES IS',I5,4X,
     2    'DOES NOT MATCH PREVIOUS: ',I5)
  203 FORMAT('0',7X,'*** UNIT',2X,I4,5X,'PARTICLE SYMBOL NUMBER',
     1    I4,2X,'IS',2X,A8,4X,'DOES NOT MATCH PREVIOUS: ',2X,A8)
C
      READ (LUDAT,END=4000,ERR=5000) INTEMP
      CALL MKCHAR (INTEMP(1),LDATE)
      CALL MKCHAR (INTEMP(3),LTIME)
      DO 800 I=1,5
         CALL MKCHAR (INTEMP(64+I),JOBID(I))
  800 CONTINUE
      DO 900 I=1,15
         CALL MKCHAR (INTEMP(69+I),IDENT(I))
  900 CONTINUE
      WRITE (6,201) LDATE, LTIME, JOBID, MAXRUN, IDENT
      IF (ITER.LE.1) GO TO 3000
         IF (INTEMP(7).EQ.NTYPE) GO TO 1000
            WRITE (6,202) LUDAT, INTEMP(7), NTYPE
            GO TO 5000
 1000    CONTINUE
         DO 2000 I=1,NTYPE
            IF (INTEMP(48+I).EQ.LTYPE(I)) GO TO 2000
               CALL MKCHAR (INTEMP(48+I),NEWTYP)
               CALL MKCHAR (INREC(48+I),OLDTYP)
               WRITE (6,203) LUDAT, I, NEWTYP, OLDTYP
               GO TO 5000
 2000       CONTINUE
 3000 CONTINUE
      DO 3500 I=1,128
         INREC(I) = INTEMP(I)
 3500 CONTINUE
      KODE = 1
      RETURN
C
 4000 CONTINUE
      KODE = 2
      RETURN
C
 5000 CONTINUE
      KODE = 3
      RETURN
      END
      SUBROUTINE INPTB (CHKODE)
C
```

```
C  12 MARCH 1986
C
C  INPTB reads parameter records 10-
C
C  Values of CHKODE returned by INPTB:
C       'OK       ' if data records were read sucessfully.
C       'EOF      ' if end-of-file read (all records did not exist).
C       'ERROR    ' if error on attempt to read record.
C
C
      CHARACTER*5 FPRIM, RPRIM, FTARG, RTARG, INFO
      CHARACTER CHKODE*8, CHLINE*81, CHMU*2
C           Following arrays provide for 40 bins, but only 20 are
C           allowed in some other subprograms.
      COMMON /BINS/ POLDEG(0:40), GMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
      COMMON /OUTSWT/ FPRIM, RPRIM, FTARG, RTARG, INFO
      COMMON /SPECS/ XNORML
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C
C         SAVE values.
      SAVE
C
      DATA MXBINS /40/
C
  100 FORMAT(A80)
  101 FORMAT(10X, I2, 1X, A2)
  102 FORMAT(10X, 7F10.0)
  103 FORMAT(10X, F10.0)
C
      CHKODE = 'OK '
C
C                       ********************************
C                       *                              *
C                       *   THETA (POLAR ANGLE) BINS   *
C                       *                              *
C                       ********************************
C
C         Read number of mu (cosine theta) bins to use and data type
C         specifier (CHMU).   If CHMU='MU', read bin boundaries in
C         units of mu and set array POLDEG to zero.   If CHMU is not
C         'MU', bin boundaries are in degrees, read into array POLDEG.
C
C                                                     *** Record 10 ***
      READ (LUSPEC, 101, END=3000, ERR=4000) NMUBIN, CHMU
      IF (CHMU.EQ.'MU') THEN
C                                                  *** Record(s) 11 ***
         READ (LUSPEC, 102, END=3000, ERR=4000) (GMU(I), I=0, NMUBIN)
         DO 1000 I=0, MXBINS
            POLDEG(I) = 0.0
 1000    CONTINUE
      ELSE
         READ (LUSPEC, 102, END=3000, ERR=4000) (POLDEG(I), I=0, NMUBIN)
      END IF
C
C                      **********************************
C                      *                                *
C                      *   PHI (AZIMUTHAL ANGLE) BINS   *
C                      *                                *
C                      **********************************
```

```
C
C         Read number of azimuthal angle bins and bin boundaries.
C
C                                                       *** Record 12 ***
      READ (LUSPEC,101,END=3000,ERR=4000) NAZBIN
C                                                    *** Record(s) 13 ***
      READ (LUSPEC,102,END=3000,ERR=4000) (AZMDEG(I),I=0,NAZBIN)
C
C                       ***********************
C                       *                     *
C                       *   ROUNDING FACTOR   *
C                       *                     *
C                       ***********************
C
C                                                       *** Record 14 ***
      READ (LUSPEC,103,END=3000,ERR=4000) XNORML
C
C                  ***********************************
C                  *                                 *
C                  *   OUTPUT ELIMINATION SWITCHES   *
C                  *                                 *
C                  ***********************************
C
C        Set defaults, specifying all output is desired.
C
      FPRIM = 'FPRIM'
      RPRIM = 'RPRIM'
      FTARG = 'FTARG'
      RTARG = 'RTARG'
      INFO  = 'INFO '
C
C        Read record from specification file and decode data in columns
C        11-80.
C
C                                                       *** Record 15 ***
      READ (LUSPEC,100,END=3000,ERR=4000) CHLINE(1:80)
      CHLINE(81:81) = ' '
      DO 2000 I=11,77
         IF (CHLINE(I:(I+4)).EQ.'FPRIM') THEN
            FPRIM = '     '
         ELSE IF (CHLINE(I:(I+4)).EQ.'RPRIM') THEN
            RPRIM = '     '
         ELSE IF (CHLINE(I:(I+4)).EQ.'FTARG') THEN
            FTARG = '     '
         ELSE IF (CHLINE(I:(I+4)).EQ.'RTARG') THEN
            RTARG = '     '
         ELSE IF (CHLINE(I:(I+3)).EQ.'INFO') THEN
            INFO = '     '
         END IF
 2000 CONTINUE
      GO TO 5000
 3000 CONTINUE
      CHKODE = 'EOF'
      GO TO 5000
 4000 CONTINUE
      CHKODE = 'ERROR'
 5000 CONTINUE
      RETURN
      END
      SUBROUTINE MKCHAR (INWORD, CHOUT)
```

```
C
C  23 JANUARY 1986   11:09
C
C  MKCHAR CONVERTS THE VALUE IN INWORD, BYTE-BY-BYTE, INTO THE CHARACTER
C  VARIABLE CHOUT.
C
      CHARACTER*8 CHOUT
      INTEGER INWORD
C
      DO 1000 I=1,7
         CHOUT(I:I) = CHAR(AND(X'00000000000000FF',SHIFT(INWORD,8*I)))
 1000 CONTINUE
      CHOUT(8:8) = CHAR(AND(X'00000000000000FF',INWORD))
      RETURN
      END
      SUBROUTINE OUPTAA
C
C  22 MAY 1986
C
C  OUPTAA PRINTS DISTRIBUTION TABLES
C
      CHARACTER*5 FPRIM, RPRIM, FTARG, RTARG
C
      DIMENSION LABSRF(3,2)
C
      COMMON /BINS/ POLDEG(0:40), QMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
      COMMON /FORFAC/ FACDIV(11,2)
      COMMON /FSTREC/ ADUMMY(5), MAXRUN, NTYPE, BDUMMY(41), LTYPE(10),
     1                DDUMMY(70)
      COMMON /MXDMNS/ MXSURF, MXTYPE, MXPOLR, MXAZM
      COMMON /MORBNS/ XMUMID(20), PHIMID(40)
      COMMON /OUTSWT/ FPRIM, RPRIM, FTARG, RTARG
      COMMON /READIN/ SBND(11), WIDTH(11), DEPBIN, DEPMAX
      COMMON /SPECS/ XNORML
      COMMON /TABLES/ NENERG(100,11,2), NYIELD(21,10,2),
     1                NDEPTH(21,10,2), NRMLZD(21,21,11,2), FACNRM(11,2),
     2                NANGLE(21,21,11,2), KSUM(10,2), KSUMSQ(10,2),
     3                NPRYMS
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C
      SAVE
C
C                Factor for converting degrees to radians
      DATA DEGRAD /0.0174532925/
      DATA LABSRF /' REF','LECT','ION ','TRAN','SMIS','SION'/
C             MAXIMUM NUMBER OF BINS IN 'TABLES' COMMON BLOCK ARRAY
C             NENERG
      DATA MXMU /21/, MXPHI /21/, NBINLM /100/
C
  201 FORMAT('1',43X,'PRIMARY PARTICLE ',3A4,' COEFFICIENTS')
  202 FORMAT('1',45X,3A4,' SPUTTERING OF ',A2,I3,' ATOMS'/
     1  1H ,43X,'SURFACE BINDING ENERGY (E.V.)',G14.6)
  203 FORMAT(1H0,52X,'DISTRIBUTION OF YIELD VALUES'/
     1  1H ,8X,'YIELD',8X,'0',20I5,'+')
  204 FORMAT(1H ,8X,'FREQUENCY',21I5)
  205 FORMAT(1H0,25X,'DISTRIBUTION OF ORIGINAL DEPTHS OF SPUTTERED',
     1     ' ATOMS (CHANNEL WIDTH',G13.6,')'/
     2  1H ,8X,'DEPTH',4X,21I5,'+')
  206 FORMAT(1H0,43X,'DISTRIBUTION OF EJECTED PARTICLE DIRECTIONS'/
     1  ' ',39X,'POLAR ANGLE (ACROSS)'/
     2  1H ,39X,'CHANNEL WIDTH (DOWN), AZIMUTHAL ANGLE (DEGREES)',
     3   G13.6/
     4  ' ',10X,'CHAN',20I5,2X,'TOTALS'/ (' ',10X,I3,1X,20I5,I7))
 207 FORMAT(1H0,13X,'PRIMARY PARTICLES',7X,'MEAN YIELD',14X,'ERROR'/
     1  1H ,18X,I5,13X,F10.2,10X,F10.2)
 208 FORMAT('0','Totals (*dPhi)',20I5,I7)
 209 FORMAT('1',43X,'DISTRIBUTION OF EJECTED PARTICLE DIRECTIONS'/
     1  'OParticle Count/(dMu*dPhi), normalized by factor (',G11.4,
     2   ' * ',G11.4,'/',G11.4,') = ',G11.4/
     3  5X,'Channel',1X,20I5/
     4  9X,'dMu',3X,20(1X,F4.2)/
     5  2X,'Phi',5X,'Mu',1X,21(1X,F4.2)/
     6  ' Chan Rad  Deg I',20(4X,'I'),1X,'Totals'/
     7  4X,F5.1,F6.1,102X,'(*dMu)'/
     8  (' ',I3,F5.1,F6.1,20I5,I7))
 210 FORMAT(' ',14X,21('I    ') / ' POL.ANG.DEG',21F5.1/
     1  ' POL.ANG.RAD',21F5.1)
 211 FORMAT('1'/'0',43X,'DISTRIBUTION OF EJECTED PARTICLE DIRECTIONS')
 212 FORMAT('0',57X,'PARTICLE COUNTS')
 213 FORMAT('0',14X,'Bin',2X,20I5 / 15X,'dMu',3X,20F5.2 /
     1  16X,'Mu',3X,20F5.2 /
     2  9X,'dPhi',3X,'Phi',1X,20(4X,'I'),4X,'Row' /
     3  3X,'Bin',1X,2(1X,'(Deg)'),1X,20(4X,'I'),4X,'Sums' //
     4  (4X,I2,1X,F5.1,3X,I3,3X,20I5,I7 /
     5  4(4X,I2,1X,F5.1,3X,I3,3X,20I5,I7/)))
 214 FORMAT(8X,'Column Sums',2X,20I5,I7 /
     1  '0',7X,'Theta (Deg)',2X,20I5 /
     2  7X,'dTheta (Deg)',2X,20F5.1)
C
      DELAZM = 360.0 / FLOAT(NAZBIN)
      AVYELD = 0.0
      ERROR = 0.0
      DO 7000 KSURF=1,2
         IF (((KSURF.EQ.1).AND.(FPRIM.EQ.'FPRIM')).OR.
     1      ((KSURF.EQ.2).AND.(RPRIM.EQ.'RPRIM'))) THEN
            DO 1000 I=1,NBINLM
               IF (NENERG(I,11,KSURF).NE.0) GO TO 2000
 1000       CONTINUE
         END IF
         GO TO 3000
 2000    CONTINUE
C                                  PRINT PRIMARY PARTICLE TABLES
         PRINT 201, (LABSRF(I,KSURF),I=1,3)
         KTYPE = 11
         CALL OUPTBB (KTYPE, KSURF)
         PRINT 206, DELAZM, (I,I=1,20),
     1      ((J, (NANGLE(K,J,11,KSURF),K=1,21)),J=1,NAZBIN)
         IF (NAZBIN.GT.1) WRITE (6,208) (NANGLE(J,21,11,KSURF),J=1,20)
         IF (KSURF.EQ.1) CALL OUPTCC (KTYPE,DEPMAX)
         CALL ANGPLT(KTYPE, KSURF, AVYELD, ERROR)
 3000    CONTINUE
         IF (((KSURF.EQ.1).AND.(FTARG.EQ.'FTARG')).OR.
     1      ((KSURF.EQ.2).AND.(RTARG.EQ.'RTARG'))) THEN
         DO 6000 KTYPE=1,NTYPE
            DO 4000 I=1,NBINLM
               IF (NENERG(I,KTYPE,KSURF).GT.0) GO TO 5000
 4000          CONTINUE
            GO TO 6000
 5000       CONTINUE
```

```
C                                  PRINT SECONDARY PARTICLE TABLES
            PRINT 202, (LABSRF(I,KSURF), I=1,3), LTYPE(KTYPE), KTYPE,
     1         SBND(KTYPE)
            SUM = FLOAT(KSUM(KTYPE,KSURF))
            PRIMES = FLOAT(NPRYMS)
            AVYELD = SUM / PRIMES
            IF (NPRYMS.LE.1) ERROR = 0.0
            IF (NPRYMS.GT.1) ERROR = SQRT((FLOAT(KSUMSQ(KTYPE,KSURF))-
     1         ((SUM*SUM)/PRIMES))/FLOAT(NPRYMS*(NPRYMS-1)))
            PRINT 207, NPRYMS, AVYELD, ERROR
            CALL OUPTBB (KTYPE, KSURF)
            PRINT 203, (I,I=1,20)
            PRINT 204, (NYIELD(I,KTYPE,KSURF),I=1,21)
            PRINT 205, DEPBIN, (I,I=1,21)
            PRINT 204, (NDEPTH(I,KTYPE,KSURF),I=1,21)
            IF (KSURF.EQ.1) CALL OUPTCC (KTYPE,DEPMAX)
            IF (NANGLE(MXPOLR,MXAZM,KTYPE,KSURF).GT.0) THEN
               WRITE (LUPRT,211)
               WRITE (LUPRT,212)
               WRITE (LUPRT,213) (I,I=1,NMUBIN),(DMU(J),J=1,NMUBIN),
     1            ((0.5*(GMU(K)+GMU(K-1))),K=1,NMUBIN),
     2            ((L,(AZMDEG(L)-AZMDEG(L-1)),
     3            (NINT(0.5*(AZMDEG(L)+AZMDEG(L-1)))),
     4            (NANGLE(M,L,KTYPE,KSURF),M=1,NMUBIN),
     5            (NANGLE(MXPOLR,L,KTYPE,KSURF))),L=1,NAZBIN)
               WRITE (LUPRT,214)
     1            ((NANGLE(I,MXAZM,KTYPE,KSURF)),I=1,NMUBIN),
     2            NANGLE(MXPOLR,MXAZM,KTYPE,KSURF),
     3            ((NINT(0.5*(POLDEG(J)+POLDEG(J-1)))),J=1,NMUBIN),
     4            ((POLDEG(K-1)-POLDEG(K)),K=1,NMUBIN)
            END IF
            WRITE (LUPRT,209) AVYELD, XNORML, FACDIV(KTYPE,KSURF),
     1         FACNRM(KTYPE,KSURF), (I,I=1,NMUBIN),
     2         (DMU(J),J=1,NMUBIN), (GMU(K),K=0,NMUBIN),
     2         AZMRAD(0), AZMDEG(0), ((L,AZMRAD(L),AZMDEG(L),
     3         (NRMLZD(M,L,KTYPE,KSURF),M=1,NMUBIN),
     4         NRMLZD(MXMU,L,KTYPE,KSURF)),L=1,NAZBIN)
            IF (NAZBIN.GT.1) WRITE (6,208) (NRMLZD(J,MXPHI,KTYPE,KSURF),
     1         J=1,NMUBIN)
            WRITE (6,210) (POLDEG(I),I=0,NMUBIN),
     1         ((DEGRAD*POLDEG(I)),I=0,NMUBIN)
            CALL ANGPLT(KTYPE, KSURF, AVYELD, ERROR)
 6000       CONTINUE
         END IF
 7000    CONTINUE
      RETURN
      END
      SUBROUTINE OUPTBB (KTYPE, KSURF)
C
C  4 MARCH 1986
C
C  OUPTBB PRINTS ENERGY DISTRIBUTION TABLE
C
C                 *****   CALLING PARAMETERS   *****
C
C  KTYPE  (INTEGER, PASSED) PARTICLE TYPE (1-10 SECONDARY, 11 PRIMARY)
C  KSURF  (INTEGER, PASSED) 1 = TARGET FRONT SURFACE, 2 = BACK SURFACE
C
C                  *****   VARIABLE DECLARATIONS   *****
C
```

```
      INTEGER  I, J, KOUNT, KSURF, KTYPE, NBIN, NBINLM, NENERG, NUM
      REAL     ADUMMY, WIDTH
C
      DIMENSION KOUNT(20), NBIN(20)
C
C              ADUMMY IS NOT USED IN THIS SUBPROGRAM
      COMMON /READIN/ ADUMMY(11), WIDTH(11)
C              ENERGY (NENERG), YIELD (NYIELD), DEPTH (NDEPTH), AND
C                       ANGULAR (NANGLE) DISTRIBUTION TABLES.
C              KSUM, KSUMSQ = SUM AND SUM OF SQUARES OF YIELD FOR
C                       CALCULATING MEAN YIELD AND STANDARD DEVIATION.
C              NPRYMS = TOTAL NUMBER OF PRIMARY PARTICLES.
      COMMON /TABLES/ NENERG(100,11,2), NYIELD(21,10,2),
     1                NDEPTH(21,10,2), XANGLE(21,21,11,2), FACNRM(11,2),
     2                NANGLE(21,21,11,2), KSUM(10,2), KSUMSQ(10,2),
     3                NPRYMS
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
      SAVE
C
C                 ENERGY BIN DIMENSION IN ARRAY NENERG
      DATA NBINLM /100/
C
  200 FORMAT(1H0,33X,'EJECTED PARTICLE ENERGY SPECTRUM (',G13.6,
     1     ' E.V. PER CHANNEL)'/ 1H0,8X,'CHANNEL',2X,20I5)
  201 FORMAT(1H0,8X,'CHANNEL',2X,20(1X,A4))
  202 FORMAT(1H ,8X,'FREQUENCY',20I5)
  401 FORMAT(I4)
  402 FORMAT(I3,'+')
C
C              Print counts for first 20 channels
C
      PRINT 200, WIDTH(KTYPE), (I,I=1,20)
      PRINT 202, (NENERG(I,KTYPE,KSURF), I=1,20)
C
C              Beyond channel 20 print only non-zero channels
C              Last channel is count of "all greater than next-to-last"
C
      NUM = 0
      DO 2000 I=21,NBINLM
         IF (NENERG(I,KTYPE,KSURF).LE.0) GO TO 1000
            NUM = NUM + 1
            KOUNT(NUM) = NENERG(I,KTYPE,KSURF)
            IF (I.LT.NBINLM) ENCODE(4,401,NBIN(NUM))I
            IF (I.GE.NBINLM) ENCODE(4,402,NBIN(NUM))I
 1000    CONTINUE
         IF ((NUM.LE.0).OR.((NUM.LT.20).AND.(I.LT.NBINLM))) GO TO 2000
            PRINT 201, (NBIN(J),J=1,NUM)
            PRINT 202, (KOUNT(J),J=1,NUM)
            NUM = 0
 2000    CONTINUE
      RETURN
      END
      SUBROUTINE OUPTCC (KTYPE, DEPMAX)
C
C  23 JANUARY 1986
C
C  OUPTCC PRINTS TABLES OF MAXIMUM PARTICLE DEPTHS, NUMBER OF
C  COLLISIONS, AND NUMBERS AND LENGTHS OF COLLISION SEQUENCES ADDED TO
C  OUTPUT FOR PROGRAM MARLOW VERSION OF 6 FEBRUARY 1982
C
```

```
      INTEGER  I, J, K, KTYPE, L, LENSEQ, MXDEP, NCOL, NCOLIS, NROW
      REAL     DEPMAX
C             COLUMN AND ROW TOTALS FOR COLLISION SEQUENCE TABLE
      DIMENSION NCOL(15), NROW(5)
C             FRONT-SURFACE-SPUTTERED PARTICLE DATA
C             MXDEP = MAXIMUM DEPTH DISTRIBUTION TABLE
C             LENSEQ = NUMBER AND LENGTH OF REPLACEMENT SEQUENCES TABLE
C             NCOLIS = NUMBER OF COLLISIONS FROM DEEPEST POINT TABLE.
C                      FIRST INDEX RUNS 1-21 FOR BIN VALUES 0-20.
      COMMON /FRSURF/ MXDEP(21,11), LENSEQ(15,5,11), NCOLIS(21,11)
      SAVE
C
  201 FORMAT(1H0,26X,'DISTRIBUTION OF MAXIMUM DEPTHS OF SPUTTERED ',
     1     'ATOMS (CHANNEL WIDTH',G13.4,')'/
     2  1H ,8X,'MAX.DEPTH',1X,21I5,'+')
  202 FORMAT(1H ,8X,'FREQUENCY ',21I5)
  203 FORMAT(/1H0,9X,'NO. OF',25X,'NUMBER OF REPLACEMENTS IN SEQUENCE',
     1     22X,'NUMBER OF'/
     2  1H ,8X,'SEQUENCES',1X,15I5,3X,'PARTICLES'//
     3  5(1H ,12X,I1,5X,15I5,5X,I5/),
     4  1H0,10X,'TOTALS',2X,15I5)
  204 FORMAT(1H0,39X,
     1     'NUMBER OF COLLISIONS FROM MAXIMUM DEPTH TO SURFACE'/
     2  1H ,8X,'COLLISIONS',4X,'0',20I5,'+')
C
      PRINT 201, DEPMAX, (I,I=1,21)
      PRINT 202, (MXDEP(I,KTYPE),I=1,21)
      DO 900 I=1,5
         NROW(I) = 0
  900 CONTINUE
      DO 1000 I=1,5
         DO 1000 J=1,15
            NROW(I) = NROW(I) + LENSEQ(J,I,KTYPE)
 1000       CONTINUE
C                          TO GET NUMBER OF PARTICLES PER ROW, MUST
C                          DIVIDE NROW(I) BY NUMBER OF SEQUENCES PER
C                          PARTICLE IN ITH ROW.
      DO 2000 I=2,5
         NROW(I) = NROW(I)/I
 2000    CONTINUE
      DO 2500 I=1,15
         NCOL(I) = 0
 2500 CONTINUE
      DO 3000 I=1,15
         DO 3000 J=1,5
            NCOL(I) = NCOL(I) + LENSEQ(I,J,KTYPE)
 3000       CONTINUE
      PRINT 203, (I,I=1,15), (J,(LENSEQ(K,J,KTYPE),K=1,15),
     1               NROW(J),J=1,5), (NCOL(L),L=1,15)
      PRINT 204, (I,I=1,20)
      PRINT 202, (NCOLIS(J,KTYPE),J=1,21)
      RETURN
      END
      SUBROUTINE PRTCLS (ISET, JFYL, LUDAT, KODE)
C
C  12 MARCH 1986
C
C  One MARLOW 'POPDAT' file (except file's first record) is processed
C  on each call to PRTCLS.
C  PRTCLS CALCULATES NEW VALUES FOR EK (PARTICLE ENERGY) AND RCOS3
```

```
C  ("DIRECTION" COSINE FROM NORMAL TO TARGET SURFACE) AND COUNTS
C  PARTICLES FOR ENERGY (NENERG), YIELD (NYIELD), DEPTH (NDEPTH),
C  AND POLAR AND AZIMUTHAL ANGLE (XANGLE) DISTRIBUTIONS.
C  THIS VERSION OF PRTCLS (FOR PROGRAM MARPOP VERSION 2.0) READS AND
C  PROCESSES RECORDS FROM FILE FT17F001 WRITTEN BY PROGRAM MARLOW
C  VERSION OF 6 FEBRUARY 1982 AND CALLS SUBROUTINE FRONT TO TABULATE
C  MAXIMUM DEPTH DISTRIBUTION AND NUMBERS AND LENGTHS OF DISPLACEMENT
C  SEQUENCES.
C
C                   *****   CALLING PARAMETERS   *****
C
C  ISET    (integer, passed)
C  JFYL    (integer, passed) Sequence number of current POPDAT data file.
C  LUDAT   (integer, passed) Logical unit from which POPDAT file is to be
C          read.
C  KODE (INTEGER, RETURNED)  VALUES OF KODE RETURNED BY PRTCLS:
C       1  RECORDS READ AND DATA PROCESSED OK.
C       2  UNEXPECTED END-OF-FILE, NO DATA.
C       3  UNEXPECTED END-OF-FILE, PARTIAL DATA READ AND PROCESSED
C       4  READ ERROR, NO DATA PROCESSED
C       5  READ ERROR, PARTIAL DATA READ AND PROCESSED.
C       6  MAXIMUM NUMBER OF RECORDS READ AND PROCESSED.  NO EXIT FLAG.
C
C              TEMPORARY STORAGE FOR PARTICLE YIELD COUNTS
      DIMENSION KYELD(10000,10,2)
      COMMON /BINS/ POLDEG(0:40), QMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
C              STORAGE FOR RECORD READ FROM FILE FT17F001
      DIMENSION INREC(8,16), XINREC(8,16)
C              ADUMMY = ARRAY NOT USED IN THIS ROUTINE
C              MAXRUN = NUMBER OF PRIMARY PARTICLES IN MARLOW RUN
C              NTYPE  = NUMBER OF TARGET PARTICLE TYPES
      COMMON /FSTREC/ ADUMMY(5), MAXRUN, NTYPE, BDUMMY(121)
C              LIMITS ON NRUN - NUMBER OF PRIMARY PARTICLES TO PROCESS
C              FROM EACH FT17FXXX FILE
      COMMON /LIMITS/ LIMRUN(10)
C              SBND   = SURFACE BINDING ENERGIES
C              WIDTH  = WIDTH OF ENERGY DISTRIBUTION BINS FOR (1-10)
C                       SECONDARY AND (11) PRIMARY PARTICLES
C              DEPBIN = WIDTH OF BIN FOR DEPTH DISTRIBUTION
C              DEPMAX = BIN WIDTH FOR MAXIMUM DEPTH DISTRIBUTION
      COMMON /READIN/ SBND(11), WIDTH(11), DEPBIN, DEPMAX
C              ENERGY (NENERG), YIELD (NYIELD), DEPTH (NDEPTH), AND
C                       ANGULAR (NANGLE) DISTRIBUTION TABLES.
C              KSUM, KSUMSQ = SUM AND SUM OF SQUARES OF YIELD FOR
C                       CALCULATING MEAN YIELD AND STANDARD DEVIATION.
C              NPRYMS = TOTAL NUMBER OF PRIMARY PARTICLES.
      COMMON /TABLES/ NENERG(100,11,2), NYIELD(21,10,2),
     1                NDEPTH(21,10,2), XANGLE(21,21,11,2), FACNRM(11,2),
     2                NANGLE(21,21,11,2), KSUM(10,2), KSUMSQ(10,2),
     3                NPRYMS
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
      EQUIVALENCE (INREC(1,1),XINREC(1,1))
C
C              Save all values on RETURN from this subroutine.
      SAVE
C
C              MAXIMUM NUMBER OF PRIMARY PARTICLES PERMITTED BY ARRAY
C              KYELD
      DATA MXPTCL /10000/
```

```
C               MAXIMUM NUMBER OF BINS IN 'TABLES' COMMON BLOCK ARRAY
C               NENERG
      DATA NBINLM /100/
C               TWOPI  = 2*PI
      DATA PI /3.141592653590/, TWOPI /6.283185307179/
C
C
  201 FORMAT('0',7X,'***',I5,' PRIMARY PARTICLES EXCEEDS LIMIT OF',I5,
     1    5X,'YIELD FROM EXCESS WILL NOT BE COUNTED')
  202 FORMAT('0',7X,'*** NUMBER OF AZIMUTHAL ANGLE BINS, ',I5,2X,
     1    'IS TOO LARGE.  CHANGING TO MAXIMUM =',I5)
  203 FORMAT(' BAD COSINE IN ROUTINE PRTCLS=',G20.10,3X,'OFF BY',G20.10,
     1    ', RESET TO 1 OR -1'/
     2 5X,'FILE',I3,', RECORD',I7,', PARTICLE',I3,'DIRECTION COSINES',
     3   2(G20.10,', '),G20.10)
  204 FORMAT(' MU OUT-OF-BOUNDS, SPECIFICATION SET',I3,
     1    ', LOGICAL UNIT',I3,', RECORD',I8,', PARTICLE',I3,', OFF BY',
     2    G12.4,', CHANGED TO',F3.0)
C
C               Zero arrays before processing data from first POPDAT file.
C
      IF (JFYL.LE.1) THEN
         DO 910 I=1,2
            DO 910 J=1,11
               DO 910 K=1,100
                  NENERG(K,J,I) = 0
  910    CONTINUE
         DO 920 I=1,2
            DO 920 J=1,10
               DO 920 K=1,21
                  NYIELD(K,J,I) = 0
  920    CONTINUE
         DO 930 I=1,2
            DO 930 J=1,10
               DO 930 K=1,21
                  NDEPTH(K,J,I) = 0
  930    CONTINUE
         DO 960 I=1,2
            DO 960 J=1,10
               KSUM(J,I) = 0
  960    CONTINUE
         DO 970 I=1,2
            DO 970 J=1,10
               KSUMSQ(J,I) = 0
  970    CONTINUE
         DO 985 I=1,2
            DO 985 J=1,11
               DO 985 K=1,21
                  DO 985 L=1,21
                     NANGLE(L,K,J,I) = 0
  985    CONTINUE
         NPRYMS = 0
C
C               Initialize subroutine FRONT.
C
         CALL FRINIT
      END IF
C
C               Set binding energy for primary particle and zero array
C               KYELD before processing data from each POPDAT file.
```

```
C
      SBND(11) = 0.0
      DO 990 I=1,2
         DO 990 J=1,10
            DO 990 K=1,10000
               KYELD(K,J,I) = 0
  990 CONTINUE
C
C             Set MAXRUN, the number of particles to be processed from
C             current POPDAT file.
C
      IF ((LIMRUN(JFYL).GT.0).AND.(LIMRUN(JFYL).LT.MAXRUN))
     1   MAXRUN = LIMRUN(JFYL)
      IF (MAXRUN.LE.MXPTCL) GO TO 2000
         PRINT 201, MAXRUN, MXPTCL
         MAXRUN = MXPTCL
 2000 CONTINUE
      AZMBIN = TWOPI / FLOAT(NAZBIN)
C
C             Each pass through DO 11000 I loop processes one record
C             from POPDAT file.
C
      DO 11000 I=1,1000000
         READ (LUDAT,END=12000,ERR=14000) INREC
C
C             Each pass through DO 10000 J loop processes data for one
C             particle.
C
         DO 10000 J=1,16
C
C             INREC(8,J) (Marlowe variable NRUN) out-of-bounds is flag
C             that last particle has been processed.
C
            IF ((INREC(8,J).LE.0).OR.(INREC(8,J).GT.MAXRUN)) GO TO 16000
C                                    CALCULATE EK CORRECTED FOR SURFACE
C                                    BINDING ENERGY AND NEW Z-DIRECTION
C                                    COSINE.  INCREMENT NENERG BIN IF
C                                    PARTICLE ESCAPES.
C
C             Extract particle type (11=primary)
C
            KTYPE = MOD(INREC(7,J),256)
            IF (MOD(INREC(6,J),2).EQ.1) KTYPE=11
C
C             Recalculate particle energy as (EK*RCOS3**2) and process
C             if particle escapes (new energy > SBND)
C
            TEMP = XINREC(1,J)*XINREC(4,J)*XINREC(4,J) - SBND(KTYPE)
            IF (TEMP.LE.0.0) GO TO 10000
C
C             Calculate new direction cosine (ZMU) of escaped particle.
C             If calculated ZMU is invalid cosine, adjust it and write
C             message on file INFO.
C
            ZMU = SQRT(TEMP/(XINREC(1,J)-SBND(KTYPE)))
            IF ((ZMU.GT.1.0).OR.(ZMU.LT.0.0)) THEN
               IF (ZMU.GT.1.0) THEN
                  ERRSIZ = ZMU - 1.0
                  ZMU = 1.0
               ELSE
            ERRSIZ = ZMU
            ZMU = 0.0
         END IF
         WRITE (LUINFO,204) ISET, LUDAT, I, J, ERRSIZ, ZMU
      END IF
C
C       Reset XINREC(1,J) to energy of escaped particle.
C
      XINREC(1,J) = XINREC(1,J) - SBND(KTYPE)
C
C       Extract KARMA from INREC(6,j) and divide by 10 to
C       determine surface from which particle escapes (KSURF=1
C       for front, 2 for rear).   KARMA is expected to be in
C       range 10-29.   Increment energy distribution count.
C
      KSURF = MOD(INREC(6,J),65536)/10
      NENBIN = 1 + IFIX(XINREC(1,J)/WIDTH(KTYPE))
      IF (NENBIN.GT.NBINLM) NENBIN=NBINLM
      NENERG(NENBIN,KTYPE,KSURF) = NENERG(NENBIN,KTYPE,KSURF) + 1
C
C       If secondary particle, increment yield count, extract
C       original depth of particle, and increment depth
C       distribution table.
C
      IF (KTYPE.EQ.11) GO TO 5000
         KYELD(INREC(8,J),KTYPE,KSURF) =
     1         KYELD(INREC(8,J),KTYPE,KSURF) + 1
         KDEP = 1 + IFIX(AMOD(XINREC(5,J),100.0)/DEPBIN)
         IF (KDEP.GT.21) KDEP=21
         NDEPTH(KDEP,KTYPE,KSURF) = NDEPTH(KDEP,KTYPE,KSURF) + 1
 5000 CONTINUE
C
C         Calculate indices of mu and phi bins and increment
C         angular distribution table
C
      DO 6000 KMU=0,NMUBIN
         IF (ZMU.GT.GMU(KMU)) GO TO 6000
         GO TO 7000
 6000 CONTINUE
      KMU = NMUBIN + 1
 7000 CONTINUE
C                             FIND QUADRANT OF AZIMUTHAL ANGLE AND
C                             CALCULATE ANGLE IN RADIANS
      COSPHI = XINREC(2,J)/SQRT((XINREC(2,J)*XINREC(2,J))+
     1            (XINREC(3,J)*XINREC(3,J)))
      IF ((COSPHI.GT.1.0).OR.(COSPHI.LT.(-1.0))) THEN
         DELTA = ABS(COSPHI) - 1.0
         WRITE (LUINFO,203) COSPHI, DELTA, LUDAT, I, J,
     1                    XINREC(2,J), XINREC(3,J), XINREC(4,J)
         IF (COSPHI.GT.1.0) THEN
            COSPHI = 1.0
         ELSE
            COSPHI = -1.0
         END IF
      END IF
      AZIMTH = ACOS(COSPHI)
      IF (XINREC(3,J).LT.0.0) AZIMTH = TWOPI - AZIMTH
      IF (AZIMTH.LE.AZMRAD(NAZBIN)) THEN
         PHI = AZIMTH
      ELSE
```

```
               PHI = AZIMTH - TWOPI
            END IF
            DO 8000 KPHI=0,NAZBIN
               IF (PHI.GT.AZMRAD(KPHI))GO TO 8000
               GO TO 9000
 8000       CONTINUE
            KPHI = NAZBIN + 1
 9000       CONTINUE
            IF ((KMU.GT.0).AND.(KMU.LE.NMUBIN).AND.(KPHI.GT.0).AND.
     1         (KPHI.LE.NAZBIN)) NANGLE(KMU,KPHI,KTYPE,KSURF) = 1 +
     2            NANGLE(KMU,KPHI,KTYPE,KSURF)
C                       TABULATE MAXIMUM DEPTH, NUMBER AND LENGTH OF
C                       DISPLACEMENT SEQUENCES AND NUMBER OF
C                       COLLISIONS DATA FOR PARTICLES LEAVING FRONT
C                       SURFACE
            IF (KSURF.EQ.1) CALL FRONT (KTYPE, XINREC(5,J), DEPMAX,
     1         INREC(7,J), INREC(6,J))
10000       CONTINUE
11000    CONTINUE
      KODE = 6
      GO TO 17000
12000 CONTINUE
      IF (I.GT.1) GO TO 13000
      KODE = 2
      RETURN
C
13000 CONTINUE
      KODE = 3
      GO TO 17000
14000 CONTINUE
      IF (I.GT.1) GO TO 15000
      KODE = 4
      RETURN
C
15000 CONTINUE
      KODE = 5
      GO TO 17000
16000 CONTINUE
      KODE = 1
17000 CONTINUE
C                                   ADD YIELD DATA FROM THIS FILE TO
C                                   NYIELD, KSUM, AND KSUMSQ
      DO 18000 KSURF=1,2
         DO 18000 KTYPE=1,NTYPE
            DO 18000 I=1,MAXRUN
               KSUM(KTYPE,KSURF) = KSUM(KTYPE,KSURF) +
     1                                 KYELD(I,KTYPE,KSURF)
               KSUMSQ(KTYPE,KSURF) = KSUMSQ(KTYPE,KSURF) +
     1            (KYELD(I,KTYPE,KSURF) * KYELD(I,KTYPE,KSURF))
               IBIN = KYELD(I,KTYPE,KSURF) + 1
               IF (IBIN.GT.21) IBIN=21
               NYIELD(IBIN,KTYPE,KSURF) = NYIELD(IBIN,KTYPE,KSURF) + 1
18000          CONTINUE
      NPRYMS = NPRYMS + MAXRUN
      RETURN
      END
      SUBROUTINE SUMRYZ
C
C  22 MAY 1986
C
```

```
      COMMON /BINS/ POLDEG(0:40), GMU(0:40), DMU(40), NMUBIN,
     1              AZMDEG(0:40), AZMRAD(0:40), DAZBIN(40), NAZBIN
      COMMON /FORFAC/ FACDIV(11,2)
      COMMON /FSTREC/ ADUMMY(5), MAXRUN, NTYPE, BDUMMY(121)
      COMMON /MXDMNS/ MXSURF, MXTYPE, MXPOLR, MXAZM
      COMMON /SPECS/ XNORML
      COMMON /TABLES/ NENERG(100,11,2), NYIELD(21,10,2),
     1                NDEPTH(21,10,2), NRMLZD(21,21,11,2), FACNRM(11,2),
     2                NANGLE(21,21,11,2), KSUM(10,2), KSUMSQ(10,2),
     3                NPRYMS
      COMMON /UNITS/ LUSPEC, LUPRT, LUINFO
C
      SAVE
C
C                                  CALCULATE ROW AND COLUMN SUMS FOR
C                                  PARTICLE DIRECTION TABLE
      DO 7000 KSURF=1,MXSURF
         DO 7000 KTYPE=1,MXTYPE
            IF ((KTYPE.LE.NTYPE).OR.(KTYPE.EQ.MXTYPE)) THEN
            DO 2000 I=1,NMUBIN
               NANGSM = 0
               DO 1500 J=1,NAZBIN
                  NANGSM = NANGSM + NANGLE(I,J,KTYPE,KSURF)
 1500          CONTINUE
               NANGLE(I,MXAZM,KTYPE,KSURF) = NANGSM
 2000       CONTINUE
            DO 4000 I=1,NAZBIN
               NANGSM = 0
               DO 3500 J=1,NMUBIN
                  NANGSM = NANGSM + NANGLE(J,I,KTYPE,KSURF)
 3500          CONTINUE
               NANGLE(MXPOLR,I,KTYPE,KSURF) = NANGSM
 4000       CONTINUE
               NCROSM = 0
               DO 6200 I=1,NMUBIN
                  NCROSM = NCROSM + NANGLE(I,MXAZM,KTYPE,KSURF)
 6200          CONTINUE
               NANGLE(MXPOLR,MXAZM,KTYPE,KSURF) = NCROSM
            END IF
 7000 CONTINUE
C
C               Normalize target values in array NRMLZD by factor
C               (average yield*10000) / total of column with largest Mu
C
      DO 7500 I=1,MXSURF
         DO 7500 J=1,MXTYPE
            DO 7500 K=1,MXAZM
               DO 7500 L=1,MXPOLR
                  NRMLZD(L,K,J,I) = 0
 7500 CONTINUE
      DO 9000 KSURF=1,MXSURF
         DO 9000 KTYPE=1,NTYPE
            IF (NANGLE(NMUBIN,MXAZM,KTYPE,KSURF).GT.0) THEN
               FACDIV(KTYPE,KSURF) = FLOAT(
     1            NANGLE(NMUBIN,MXAZM,KTYPE,KSURF)) / DMU(NMUBIN)
               FACTOR = (FLOAT(KSUM(KTYPE,KSURF)) / FLOAT(NPRYMS)) *
     1            XNORML / FACDIV(KTYPE,KSURF)
               FACNRM(KTYPE,KSURF) = FACTOR
               DO 8000 I=1,NAZBIN
                  DO 8000 J=1,NMUBIN
```

```
                     NRMLZD(J,I,KTYPE,KSURF) = NINT(FACTOR * FLOAT(
     1                  NANGLE(J,I,KTYPE,KSURF))/(DMU(J)*DAZBIN(I)))
 8000            CONTINUE
                 DO 8200 I=1,NAZBIN
                    NRMLZD(MXPOLR,I,KTYPE,KSURF) = NINT(FACTOR * (
     1                 FLOAT(NANGLE(MXPOLR,I,KTYPE,KSURF)) / DAZBIN(I)))
 8200            CONTINUE
                 DO 8400 I=1,NMUBIN
                    NRMLZD(I,MXAZM,KTYPE,KSURF) = NINT(FACTOR * (
     1                 FLOAT(NANGLE(I,MXAZM,KTYPE,KSURF)) / DMU(I)))
 8400            CONTINUE
              END IF
 9000 CONTINUE
      RETURN
      END
```

END

7-87

DTIC